SWTPC 6800 Parallel Interface Diagnostic PARINT-1

This parallel interface diagnostic is one designed to locate problems in the SWTPC 6800 Computer System parallel interface board, MP-L. It is assumed that before loading this program the rest of the system is functioning normally with no problems. The program itself uses 33_{10} words and is loaded within the 128 word RAM used by the MIKBUG® operating system on the MP-A Microprocessor/ System Board. A program may reside in external RAM memory simultaneously with the diagnostic loaded within the 128 word RAM, or the diagnostic may be run with no MP-M memory boards installed on the system at all. The diagnostic may be loaded either from tape or instruction by instruction using MIKBUG® starting from address AØ48, thru AØ68. The address of the parallel interface to be diagnosed is set by using MIKBUG® to load the hexadecimal address of the selected port into memory locations AØØ2 and AØØ3 with the most significant byte going into AØØ2 and the least significant byte going into AØØ3. The starting address locations of the interface ports are given below:

Port	Address in Hex	
I/O #Ø	8ØØØ	
I/O #1	8ØØ4	(reserved for control interface)
I/O #2	8ØØ8	
I/O #3	8ØØC	
I/O #4	8Ø1Ø	
I/O #5	8Ø14	
I/O #6	8Ø18	
I/O #7	8Ø1C	

Since the program counter is set when the program is initially loaded, the diagnostic is initiated as described in the "Go to User's Program" section of the Engineering Note 100. Once initiated, the program can be stopped only by depressing the "RESET" button. The program may then be re-started after resetting the program counter to AØ4A as described in the "Display contents of MPU Registers Function" section of Engineering Note 100.

The diagnostic itself works by echoing everything entered from the control terminal's keyboard back to the control terminal's display thru the parallel interface board. The control terminal remains connected to the serial control interface (I/O port #1). The normal "echo" of the control interface is first software disabled and data is then transfered thru hardwired jumpers from the parallel interface's input to output ports. Neither interrupts nor the CA1, CA2, CB1 or CB2 lines are tested. It is unlikely that the rest of the interface would check properly with only these lines inoperative.

To check the board, first make the following jumper connections on the MP-L parallel interface board I/O male connectors:

INPUT CONNECTOR		OUTPUT CONNECTOR
IØ	to	OØ
I1	to	O1
I2	to	O2
I3	to	O3
I4	to	O4
I5	to	O5
I6	to	O6
I7	to	O7

Plug the wired connectors onto their positions on the MP-L interface board and with the power off, plug the board onto the selected interface location. Power up the system and load in the diagnostic program and the address of the parallel interface. Then execute a "Go to User's Program" as described in Engineering Note 100. Now as each character is typed, it will be "echoed" back and printed on the control terminal's display. At low control interface baud rates you will notice a delay in the "echo" function which is due to the software "echo" routine. You may also notice that the "echo" doesn't work properly if you type too fast. This is normal.

If you have problems, check the data on the output connector of the interface after each key is struck. Bits Ø thru 6 on the connector should be identical to the ASCII bit pattern of the key struck with bit 7 always a zero. The data should remain latched on the output of the interface until the next key is struck.

Never install or remove the interface board when the system is powered up. Doing so is not only hazardous to the equipment, but bypasses the normal power-up sequence required by the internal registers within the 6820 integrated circuit in order to guarantee proper operation.

SWTPC Parallel Interface Diagnostic PARTINT-1

```
A002            MSB OF PARADR (PIA address)
A003            LSB OF PARADR (PIA address)
E1AC            SUBROUTINE INEEE
E1D1            SUBROUTINE OUTEEE


                   Start Loading Program At A04A

A048    A0      (Program Counter MSB)
A049    4A      (Program Counter LSB)
A04A    86      START   LDA  A    #$3A
A04B    3C
A04C    B7              STA  A    #8007
A04D    80
A04E    07
A04F    FE              LDX       PARADR
A050    A0
A051    02
A052    C6              LDA  B    #$FF
A053    FF
A054    E7              STA  B    0,X
A055    00
A056    53              COM  B
A057    E7              STA  B    Z,X
A058    02
A059    A7              STA  A    1,X
A05A    01
A05B    A7              STA  A    3,X
A05C    03
A05D    BD      LOOP1   JSR       INEEE
A05E    E1
A05F    AC
A060    A7              STA  A    0,X
A061    00
A062    A6              LDA  A    Z,X
A063    02
A064    BD              JSR       OUTEEE
A065    E1
A066    D1
A067    20              BRA       LOOP1
A068    F4
```

- END -

SWTPC 6800 TIC-TAC-TOE PROGRAM TICTAC-1

This program allows the operator to play the game of TIC-TAC-TOE at the serial control terminal. The program uses $\emptyset 6B\emptyset_{16}$ (1812_{10}) words of memory and is meant to be loaded to memory starting from location $\emptyset\emptyset 2\emptyset_{16}$. The program is actually in two parts with the first part residing in locations $\emptyset\emptyset 2\emptyset_{16}$ to $\emptyset\emptyset A2_{16}$ and the second part residing in locations $\emptyset 1\emptyset\emptyset_{16}$ to $\emptyset 6B\emptyset_{16}$. Due to the length of this program, it should be dumped to cassette or paper tape after loading. This will make it easier to load the program the next time you wish to use it.

To use the TIC-TAC program load it in step by step using the MIKBUG® memory change function or through cassette or paper tape. The program counter addresses $A\emptyset 48_{16}$ and $A\emptyset 49_{16}$ must then be set to $\emptyset 1$ and $\emptyset\emptyset$ respectively. The TIC-TAC program can now be initiated as described in the "Go to User's Program" section of Engineering Note 100. If the program was loaded correctly the computer should respond with "HELLO I AM A SWTPC 68ØØ COMPUTER I AM PROGRAMMED TO PLAY TIC-TAC-TOE..." The computer will then give some instructions, print a sample tic-tac-toe grid and will ask for "YOUR PLAY":" The numbers 1-9 designate the squares 1-9 as shown on the sample grid. When the computer says "YOUR PLAY:" simply type in the number of the square that you wish to occupy with a "Ø". The computer will then come back with "NEW=" and some number. The new number is the square that the computer chose to occupy. An updated grid is then printed. If you type in a wrong character the computer responds with "INVALID CHARACTER" or if you choose an occupied square the computer tells you so and asks for another play. When the game is finished and the computer asks if you want to play again, type a Y for yes or an N for no. The program can be exited at any time by pressing the RESET button and can be restarted by re-setting the program counter to Ø1ØØ and initializing the program as described earlier.

The original source listing of this program is contained in the Motorola M6800 User Library, however. the program was slightly modified to enable it to work with the SWTPC 6800 Mikbug® firmware and CT-1024 terminal system. These modifications did not change the internal workings of the program or its ability to run on TTY's or other type terminals.

The program sends out control character commands to home-up and erase the screen on those CT-1024 terminal systems having the CT-CA computer controlled cursor option. You should program the jumpers on your CT-CA option so a control P (10_{16}) generates a Home-Up and a Control V (16_{16}) generates an Erase to End of Frame (EOF).

If after loading your TIC-TAC program it fails to work properly go back and re-check each memory location from $\emptyset\emptyset 2\emptyset_{16}$ to $\emptyset 6B\emptyset$ to be sure that the program was loaded correctly. If it was and the program still does not work properly run the two memory diagnostics described in the Software Section of the System Documentation Notebook on the memory being used to verify that the memory being used is operational. Running the diagnostics of course will cause whatever was stored in the memory being tested to be erased, so the TIC-TAC program should be re-entered.

Mikbug® is a registered trademark of Motorola Inc.

```
0020 0A LF
0021 20 ␣
0022 31 1
0023 20 ␣
0024 21 !
0025 20 ␣
0026 32 2
0027 20 ␣
0028 21 !
0029 20 ␣
002A 33 3
002B 0D CR
002C 0A LF
002D 2D -
002E 2D -
002F 2D -
0030 21 !
0031 2D -
0032 2D -
0033 2D -
0034 21 !
0035 2D -
0036 2D -
0037 2D -
0038 0D CR
0039 0A LF
003A 20 ␣
003B 34 4
003C 20 ␣
003D 21 !
003E 20 ␣
003F 35 5
0040 20 ␣
0041 21 !
0042 20 ␣
0043 36 6
0044 0D CR
0045 0A LF
0046 2D -
0047 2D -
0048 2D -
0049 21 !
004A 2D -
004B 2D -
004C 2D -
004D 21 !
004E 2D -
004F 2D -
0050 2D -
0051 0D CR
0052 0A LF
0053 20 ␣
0054 37 7
0055 20 ␣
0056 21 !
0057 20 ␣
0058 38 8
0059 20 ␣
005A 21 !
005B 20 ␣
005C 39 9
005D 20 ␣
005E 20 ␣
005F 04 EOT
0060 00
0061 60
0062 00
0063 22
0064 00
0065 26
0066 00
0067 2A
0068 00
0069 3B
006A 00
006B 3F
006C 00
006D 43
006E 00
006F 54
0070 00
0071 58
0072 00
0073 5C
0074 00
0075 00
0076 0A
0077 00
0078 00
0079 00
007A 00
007B 00
007C 00
007D 10
007E 16
007F 7F
0080 7F
0081 4E
0082 45
0083 57
0084 3D
0085 00
0086 04
0087 00 NUL
0088 00 NUL
0089 53 S
008A 4B D
008B 49 I
008C 4C L
008D 4C L
008E 3D =
008F 39 9
0090 20 ␣
0091 04 EOT
0092 00
0093 00
0094 00
0095 00
0096 00
0097 00
0098 00
0099 05
009A BA
009B 05
009C C0
009D 05
009E C6
009F 05
00A0 CC
00A1 06
00A2 65
```

```
 1 | 2 | 3
---|---|---
 4 | 5 | 6
---|---|---
 7 | 8 | 9  ␣
```

```
0100 7E JMP Φ4E8
0101 04
0102 E8
0103 10 Φ2DLE
0104 16 Φ4SYN
0105 7F DEL
0106 7F DEL
0107 48 H
0108 45 E
0109 4C L
010A 4C L
010B 4F O
010C 20 ␣
010D 49 I
010E 20 ␣
010F 41 A
0110 4D M
0111 20 ␣
0112 41 A
0113 20 ␣
0114 53 S
0115 57 W
0116 54 T
0117 50 P
0118 43 C
0119 20 ␣
011A 36 6
011B 38 8
011C 30 0
011D 30 0
011E 20 ␣
011F 43 C
0120 4F O
0121 4D M
0122 50 P
0123 55 U
0124 54 T
0125 45 E
0126 52 R
0127 00 NUL
0128 00 NUL
0129 00 NUL
012A 00 NUL
012B 0A LF
012C 0D CR
012D 00 NUL
012E 0D CR
012F 0A LF
0130 49 I
0131 20 ␣
0132 41 A
0133 4D M
0134 20 ␣
0135 50 P
0136 52 R
0137 4F O
0138 47 G
```

```
0139 52 R
013A 41 A
013B 4D M
013C 4D M
013D 45 E
013E 44 D
013F 20 ␣
0140 54 T
0141 4F O
0142 20 ␣
0143 50 P
0144 4C L
0145 41 A
0146 59 Y
0147 0A LF
0148 0D CR
0149 54 T
014A 49 I
014B 43 C
014C 2D -
014D 54 T
014E 41 A
014F 43 C
0150 2D -
0151 54 T
0152 4F O
0153 45 E
0154 0D CR
0155 0A LF
0156 59 Y
0157 4F O
0158 55 U
0159 20 ␣
015A 41 A
015B 52 R
015C 45 E
015D 20 ␣
015E 22 "
015F 4F O
0160 22 "
0161 20 ␣
0162 41 A
0163 4E N
0164 44 D
0165 20 ␣
0166 59 Y
0167 4F O
0168 55 U
0169 20 ␣
016A 50 P
016B 4C L
016C 41 A
016D 59 Y
016E 20 ␣
016F 46 F
0170 49 I
0171 52 R
0172 53 S
0173 54 T
0174 2E .
0175 0D CR
0176 0A LF
0177 59 Y
0178 4F O
0179 55 U
017A 20 ␣
017B 50 P
017C 4C L
017D 41 A
017E 59 Y
017F 20 ␣
0180 42 B
0181 59 Y
0182 20 ␣
0183 54 T
0184 59 Y
0185 50 P
0186 49 I
0187 4E N
0188 47 G
0189 20 ␣
018A 41 A
018B 20 ␣
018C 4E N
018D 55 U
018E 4D M
018F 42 B
0190 45 E
0191 52 R
0192 20 ␣
0193 46 F
0194 52 R
0195 4F O
0196 4D M
0197 20 ␣
0198 31 1
0199 20 ␣
019A 54 T
019B 4F O
019C 20 ␣
019D 39 9
019E 20 ␣
019F 20
01A0 20
01A1 20
01A2 20
01A3 20
01A4 20
01A5 20
01A6 20
01A7 20 ␣
01A8 0D CR
01A9 0A LF
01AA 04 EOT
01AB 49 I
01AC 4E N
01AD 56 V
01AE 41 A
01AF 4C L
01B0 49 I
01B1 44 D
01B2 20 ␣
01B3 52 R
01B4 45 E
01B5 50 P
01B6 4C L
01B7 59 Y
01B8 21 !
01B9 20 ␣
01BA 50 P
01BB 4C L
01BC 45 E
01BD 41 A
01BE 53 S
01BF 45 E
01C0 20 ␣
01C1 54 T
01C2 59 Y
01C3 50 P
01C4 45 E
01C5 20 ␣
01C6 41 A
01C7 47 G
01C8 41 A
01C9 49 I
01CA 4E N
01CB 3A :
01CC 04
01CD 54 T
01CE 48 H
01CF 41 A
01D0 54 T
01D1 20 ␣
01D2 4C L
01D3 4F O
01D4 43 C
01D5 41 A
01D6 54 T
01D7 49 I
01D8 4F O
01D9 4E N
01DA 20 ␣
01DB 49 I
01DC 53 S
01DD 20 ␣
01DE 41 A
01DF 4C L
01E0 52 R
01E1 45 E
01E2 41 A
01E3 44 D
01E4 59 Y
01E5 20 ␣
01E6 54 T
01E7 41 A
01E8 4B K
01E9 45 E
01EA 4E N
01EB 2C ,
01EC 20 ␣
01ED 50 P
01EE 4C L
01EF 45 E
01F0 41 A
01F1 53 S
01F2 45 E
01F3 20 ␣
01F4 54 T
01F5 52 R
01F6 59 Y
01F7 20 ␣
01F8 41 A
01F9 47 G
01FA 41 A
01FB 49 I
01FC 4E N
01FD 2E .
01FE 20 ␣
01FF 20 ␣
0200 04
0201 59
0202 4F
0203 55
0204 52
0205 20
0206 50
0207 4C
0208 41
0209 59
020A 3A
020B 20
020C 04
020D 59
020E 4F
020F 55
0210 20
0211 57
0212 49
0213 4E
0214 21
0215 04
0216 49
0217 20
0218 57
0219 49
021A 4E
021B 21
021C 04
```

```
021D 54
021E 49
021F 45
0220 20
0221 47
0222 41
0223 4D
0224 45
0225 2E
0226 04
0227 57
0228 4F
0229 55
022A 4C
022B 44
022C 20
022D 59
022E 4F
022F 55
0230 20
0231 4C
0232 49
0233 4B
0234 45
0235 20
0236 54
0237 4F
0238 20
0239 50
023A 4C
023B 41
023C 59
023D 20
023E 41
023F 47
0240 41
0241 49
0242 4E
0243 3F
0244 20
0245 04
0246 10 02
0247 16 04
0248 00
0249 00
024A 54
024B 59
024C 50
024D 45
024E 20
024F 4E
0250 45
0251 57
0252 20
0253 53
0254 4B
0255 49
0256 4C
0257 4C
0258 3A
0259 20
025A 30
025B 2D
025C 4C
025D 4F
025E 57
025F 2C
0260 20
0261 54
0262 4F
0263 20
0264 39
0265 2D
0266 48
0267 49
0268 47
0269 48
026A 20
026B 20
026C 20
026D 20
026E 20
026F 20
0270 20
0271 20
0272 20
0273 20
0274 20
0275 20
0276 20
0277 04
0278 54
0279 48
027A 41
027B 4E
027C 4B
027D 20
027E 59
027F 4F
0280 55
0281 2E
0282 20
0283 49
0284 20
0285 48
0286 4F
0287 50
0288 45
0289 20
028A 59
028B 4F
028C 55
028D 20
028E 48
028F 41
0290 56
0291 45
0292 20
0293 0A
0294 0D
0295 45
0296 4E
0297 4A
0298 4F
0299 59
029A 45
029B 44
029C 20
029D 50
029E 4C
029F 41
02A0 59
02A1 49
02A2 4E
02A3 47
02A4 0D
02A5 0A
02A6 59
02A7 4F
02A8 55
02A9 20
02AA 4D
02AB 41
02AC 59
02AD 20
02AE 52
02AF 45
02B0 53
02B1 54
02B2 41
02B3 52
02B4 54
02B5 20
02B6 42
02B7 59
02B8 20
02B9 54
02BA 59
02BB 50
02BC 49
02BD 4E
02BE 47
02BF 20
02C0 41
02C1 4E
02C2 59
02C3 0A
02C4 0D
02C5 43
02C6 48
02C7 41
02C8 52
02C9 41
02CA 43
02CB 54
02CC 45
02CD 52
02CE 0A
02CF 0A
02D0 0A
02D1 0A
02D2 0A
02D3 0A
02D4 0A
02D5 0A
02D6 04
02D7 50
02D8 4C
02D9 45
02DA 41
02DB 53
02DC 45
02DD 20
02DE 54
02DF 59
02E0 50
02E1 45
02E2 20
02E3 22
02E4 59
02E5 22
02E6 20
02E7 4F
02E8 52
02E9 20
02EA 22
02EB 4E
02EC 22
02ED 2E
02EE 04
02EF D6 LDA B 22
02F0 22
02F1 96 LDA A 54
02F2 54
02F3 97 STA A 22
02F4 22
02F5 96 LDA A 5C
02F6 5C
02F7 97 STA A 54
02F8 54
02F9 96 LDA A 2A
02FA 2A
02FB 97 STA A 5C
02FC 5C
02FD D7 STA B 2A
02FE 2A
02FF D6 LDA B 26
0300 26
```

```
0301 96 LDA A 3E
0302 3B
0303 97 STA A 26
0304 26
0305 96 LDA A 58
0306 58
0307 97 STA A 3B
0308 3B
0309 96 LDA A 43
030A 43
030B 97 STA A 58
030C 58
030D D7 STA B 43
030E 43
030F 30 TSX
0310 EE LDX 0,X
0311 00
0312 31 INS
0313 31 INS
0314 7A DEC 0074
0315 00
0316 74
0317 26 BNE 031F
0318 06
0319 86 LDA A #04
031A 04
031B 97 STA A 97
031C 74
031D 6E JMP 2,X
031E 02
031F EE LDX 0,X
0320 00
0321 6E JMP 0,X
0322 00
0323 30 TSX
0324 9F STS 92
0325 92
0326 AE LDS 0,X
0327 00
0328 34 DES
0329 7F CLR 0096
032A 00
032B 96
032C 32 PUL A
032D 4D TST A
032E 2B BMI 0373
032F 43
0330 16 TAB
0331 C4 AND B #0F
0332 0F
0333 58 ASL B
0334 D7 STO B 95
0335 95
0336 DE
0337 94
0338 EE
0339 60
033A E6
033B 00
033C 85
033D 10
033E 27
033F 06
0340 C1
0341 58
0342 27
0343 1F
0344 20
0345 0E
0346 85
0347 20
0348 27
0349 06
034A C1
034B 4F
034C 27
034D 15
034E 20
034F 04
0350 C1
0351 20
0352 27
0353 0F
0354 85
0355 40
0356 26
0357 03
0358 7C
0359 00
035A 96
035B 74
035C 00
035D 96
035E 78
035F 00
0360 96
0361 20
0362 C9
0363 85
0364 40
0365 26
0366 C5
0367 7C
0368 00
0369 96
036A 74
036B 00
036C 96
036D 0D
036E 79
036F 00
0370 96
0371 20
0372 B9
0373 30
0374 9E
0375 92
0376 31
0377 31
0378 7D
0379 00
037A 96
037B 26
037C 02
037D 6E
037E 00
037F 84
0380 0F
0381 97
0382 95
0383 DE
0384 94
0385 EE
0386 97
0387 6E
0388 00
0389 96
038A 77
038B 84
038C 0F
038D 80
038E 05
038F 97
0390 78
0391 96
0392 75
0393 81
0394 01
0395 26
0396 2B
0397 96
0398 76
0399 81
039A 06
039B 2E
039C 12
039D 81
039E 05
039F 2E
03A0 09
03A1 2D
03A2 31
03A3 7D
03A4 00
03A5 78
03A6 2A
03A7 2C
03A8 20
03A9 05
03AA 7D
03AB 00
03AC 78
03AD 2B
03AE 25
03AF BD
03B0 03
03B1 23
03B2 22
03B3 25
03B4 82
03B5 BD
03B6 03
03B7 23
03B8 21
03B9 86
03BA BD
03BB 02
03BC EF
03BD 03
03BE AF
03BF 7E
03C0 03
03C1 BF
03C2 BD
03C3 03
03C4 23
03C5 21
03C6 62
03C7 63
03C8 21
03C9 65
03CA 69
03CB 22
03CC 65
03CD 68
03CE 88
03CF BD
03D0 02
03D1 EF
03D2 03
03D3 C2
03D4 7C
03D5 00
03D6 7B
03D7 BD
03D8 03
03D9 23
03DA 12
03DB 53
03DC 41
03DD 15
03DE 59
03DF 41
03E0 14
03E1 57
03E2 41
03E3 82
03E4 BD
```

```
03E5 03
03E6 23
03E7 11
03E8 53
03E9 42
03EA 15
03EB 58
03EC 42
03ED 84
03EE BD
03EF 03
03F0 23
03F1 11
03F2 59
03F3 45
03F4 12
03F5 58
03F6 45
03F7 86
03F8 BD
03F9 02
03FA EF
03FB 03
03FC D4
03FD 7F
03FE 00
03FF 7B
0400 96
0401 76
0402 81
0403 01
0404 2E
0405 09
0406 2D
0407 3C
0408 7D
0409 00
040A 78
040B 2A
040C 37
040D 20
040E 0F
040F 96
0410 76
0411 81
0412 02
0413 27
0414 04
0415 81
0416 05
0417 26
0418 05
0419 7D
041A 00
041B 78
041C 2B
041D 26
041E BD
041F 03
0420 23
0421 22
0422 63
0423 41
0424 25
0425 69
0426 41
0427 24
0428 67
0429 41
042A 82
042B BD
042C 03
042D 23
042E 21
042F 63
0430 42
0431 25
0432 68
0433 42
0434 84
0435 BD
0436 03
0437 23
0438 21
0439 69
043A 45
043B 22
043C 68
043D 45
043E 86
043F BD
0440 02
0441 EF
0442 04
0443 1E
0444 96
0445 75
0446 81
0447 02
0448 26
0449 5F
044A 96
044B 76
044C 81
044D 09
044E 2D
044F 09
0450 2E
0451 10
0452 7D
0453 00
0454 78
0455 2B
0456 52
0457 20
0458 09
0459 81
045A 08
045B 26
045C 4C
045D 7D
045E 00
045F 78
0460 2A
0461 47
0462 BD
0463 03
0464 23
0465 01
0466 62
0467 55
0468 67
0469 01
046A 66
046B 68
046C 49
046D 01
046E 63
046F 65
0470 01
0471 42
0472 53
0473 67
0474 01
0475 63
0476 57
0477 68
0478 82
0479 BD
047A 03
047B 23
047C 02
047D 64
047E 55
047F 02
0480 55
0481 66
0482 02
0483 61
0484 69
0485 02
0486 63
0487 67
0488 02
0489 64
048A 67
048B 02
048C 66
048D 69
048E 84
048F BD
0490 03
0491 23
0492 05
0493 62
0494 64
0495 51
0496 05
0497 62
0498 67
0499 51
049A 05
049B 63
049C 64
049D 51
049E 05
049F 61
04A0 66
04A1 68
04A2 59
04A3 86
04A4 BD
04A5 02
04A6 EF
04A7 04
04A8 62
04A9 D6
04AA 77
04AB 96
04AC 75
04AD 81
04AE 02
04AF 2E
04B0 0A
04B1 27
04B2 04
04B3 C1
04B4 20
04B5 2E
04B6 17
04B7 C1
04B8 30
04B9 2E
04BA 13
04BB 81
04BC 04
04BD 27
04BE 04
04BF C1
04C0 40
04C1 2E
04C2 0B
04C3 C1
04C4 80
04C5 2E
04C6 15
04C7 BD
04C8 03
```

```
04C9 23
04CA 05
04CB 86
04CC 20
04CD 04
04CE C1
04CF 80
04D0 2E
04D1 0A
04D2 BD
04D3 03
04D4 23
04D5 01
04D6 82
04D7 BD
04D8 02
04D9 EF
04DA 04
04DB D2
04DC BD
04DD 03
04DE 23
04DF 02
04E0 84
04E1 BD
04E2 02
04E3 EF
04E4 04
04E5 DC
04E6 20
04E7 DF
04E8 CE  LDX # 0062
04E9 00
04EA 62
04EB 86  LDAA # 31
04EC 31
04ED DF  STX 60
04EE 60
04EF EE  LDX 0,X
04F0 00
04F1 A7  STAA 0,X
04F2 00
04F3 81  CMPA # 39
04F4 39
04F5 27  BEQ 04FE
04F6 07
04F7 DE  LDX 60
04F8 60
04F9 08  INX
04FA 08  INX
04FB 4C  INC A
04FC 20  BRA 04EF
04FD EF
04FE CE  LDX # 0103
04FF 01
0500 03
0501 BD  JSR 0692
0502 06
0503 92
0504 CE  LDX #$0020
0505 00
0506 20
0507 BD  JMS 0692
0508 06
0509 92
050A CE  LDX #$62
050B 00
050C 62
050D 86  LDA A #$20
050E 20
050F C6  LDA B #$09
0510 09
0511 DF  STX 60
0512 60
0513 EE  LDX 0,X
0514 00
0515 A7  STA 0,X
0516 00
0517 5A  DEC B
0518 27  BEQ 051F
0519 06
051A DE  LDX 60
051B 60
051C 08  INX
051D 08  INX
051E 20  BRA 0511
051F F1
0520 86  LDA #$04
0521 04
0522 97  STAA 74
0523 74
0524 86  LDA #$01
0525 01
0526 97  STA A 75
0527 75
0528 7F  CLR 007B
0529 00
052A 7B
052B 7F  CLR 007C
052C 00
052D 7C
052E BD  JMS 06A1
052F 06
0530 A1
0531 96  LDAA 76
0532 76
0533 81  CMP A # 0A
0534 0A
0535 2C  BGE 0541
0536 0A
0537 8A  ORA A # 30
0538 30
0539 97  STA 8F
053A 8F
053B CE  LDX #$0097
053C 00
053D 87
053E BD  JMS 0692
053F 06
0540 92
0541 CE  LDX #$0201
0542 02
0543 01
0544 BD  JMS 0694
0545 06
0546 94
0547 7C  INC 0077
0548 00
0549 77
054A 26  BNE 054E
054B 03
054C 7C  INC 0079
054D 00
054E 79
054F B6  LDA A 8004
0550 80
0551 04
0552 01  NOP
0553 2B  BMI 0547
0554 F2
0555 BD  JMS E1AC
0556 E1
0557 AC
0558 81  CMP A #$53
0559 53
055A 26  BNE 057E
055B 22
055C 96  LDA A 75
055D 75
055E 81  CMP A 01
055F 01
0560 26  BNE 05AA
0561 48
0562 CE  LDX # 0246
0563 02
0564 46
0565 BD  JMS 0692
0566 06
0567 92
0568 BD  JMS E1AC
0569 E1
056A AC
056B 81  CMP A #$30
056C 30
056D 2D  BLT 0573
056E 04
056F 81  CMP # $39
0570 39
0571 23  BLS 0575
0572 02
0573 86  LDA A #$0A
0574 0A
0575 84  AND A # $0F
0576 0F
0577 97  STA A 76
0578 76
0579 BD  JMS 06A1
057A 06
057B A1
057C 20  BRA 0541
057D C3
057E 81  CMP A # 31
057F 31
0580 2D  BLT 05AA
0581 28
0582 81  CMP A # 39
0583 39
0584 2E  BGT 05AA
0585 24
0586 84  AND A # 0F
0587 0F
0588 48  ASL A
0589 97  STA A 95
058A 95
058B DE  LDX 94
058C 94
058D EE  LDX 60,X
058E 60
058F A6  LDA A 0,X
0590 00
0591 81  CMP A # 20
0592 20
0593 26  BNE 05B2
0594 1D
0595 86  LDA A # 4F
0596 4F
0597 A7  STA A 0,X
0598 00
0599 96  LDA A 79
059A 79
059B 84  AND A # 03
059C 03
059D 97  STA A 7A
059E 7A
059F 4C  INC A
05A0 97  STA A 74
05A1 74
05A2 BD  JMS 02EF
05A3 02
05A4 EF
05A5 05
05A6 A2
05A7 7E
05A8 03
05A9 89
05AA CE
05AB 01
05AC AB
```

```
05AD BD
05AE 06
05AF 92
05B0 20
05B1 95
05B2 CE
05B3 01
05B4 CD
05B5 BD
05B6 06
05B7 92
05B8 20
05B9 8D
05BA 86
05BB 5A
05BC 97
05BD 22
05BE 20
05BF 0F
05C0 86
05C1 5A
05C2 97
05C3 26
05C4 20
05C5 09
05C6 86
05C7 5A
05C8 97
05C9 3F
05CA 20
05CB 03
05CC 7C
05CD 00
05CE 7C
05CF 96
05D0 74
05D1 8B
05D2 03
05D3 90
05D4 7A
05D5 97
05D6 74
05D7 BD
05D8 02
05D9 EF
05DA 05
05DB D7
05DC 7D
05DD 00
05DE 7C
05DF 26
05E0 4E
05E1 C6
05E2 01
05E3 CE
05E4 00
05E5 62
05E6 DF
05E7 60
05E8 EE
05E9 00
05EA A6
05EB 00
05EC 81
05ED 5A
05EE 27
05EF 07
05F0 DE
05F1 60
05F2 5C
05F3 08
05F4 08
05F5 20
05F6 EF
05F7 86
05F8 58
05F9 A7
05FA 00
05FB CA
05FC 30
05FD D7
05FE 85
05FF CE
0600 00
0601 7D
0602 BD
0603 06
0604 92
0605 CE
0606 00
0607 20
0608 BD
0609 06
060A 92
060B 7D
060C 00
060D 7B
060E 26
060F 2D
0610 96
0611 75
0612 7C
0613 00
0614 75
0615 81
0616 04
0617 27
0618 03
0619 7E
061A 05
061B 41
061C CE
061D 00
061E 62
061F DF
0620 60
0621 EE
0622 00
0623 A6
0624 00
0625 81
0626 20
0627 27
0628 1C
0629 DE
062A 60
062B 08
062C 08
062D 20
062E F0
062F CE
0630 00
0631 20
0632 BD
0633 06
0634 92
0635 CE
0636 02
0637 0D
0638 BD
0639 06
063A 94
063B 20
063C 30
063D CE
063E 02
063F 16
0640 BD
0641 06
0642 94
0643 20
0644 28
0645 86
0646 4F
0647 A7
0648 00
0649 DF
064A 60
064B BD
064C 03
064D 23
064E 21
064F 62
0650 63
0651 21
0652 65
0653 69
0654 22
0655 65
0656 68
0657 8A
0658 BD
0659 02
065A EF
065B 06
065C 4B
065D CE
065E 02
065F 1D
0660 BD
0661 06
0662 94
0663 20
0664 08
0665 DE
0666 60
0667 86
0668 20
0669 A7
066A 00
066B 20
066C C8
066D CE
066E 02
066F 27
0670 BD
0671 06
0672 92
0673 BD
0674 E1
0675 AC
0676 81
0677 59
0678 26
0679 03
067A 7E
067B 05
067C 0A
067D 81
067E 4E
067F 26
0680 0C
0681 CE
0682 02
0683 78
0684 BD
0685 06
0686 92
0687 BD
0688 E1
0689 AC
068A 7E
068B 04
068C E8
068D CE
068E 02
068F D7
0690 20
```

```
0691 DE
0692 8D   BSR 06A1
0693 0D
0694 A6   LDA A 0,X
0695 00
0696 81   CMP A #04
0697 04
0698 27   BEQ 06A0
0699 06
069A BD   JSR E1D1      OUT EEE
069B E1
069C D1
069D 08   INX
069E 20   BRA 0694
069F F4
06A0 39   RTS
06A1 86   LDA A 0A                  LF / CR / NUL  OUT
06A2 0A
06A3 BD   JMS E1D1      OUT EEE
06A4 E1
06A5 D1
06A6 86   LDA A 0D
06A7 0D
06A8 BD   JMS E1D1      OUT EEE
06A9 E1
06AA D1
06AB 86   LDA A 00
06AC 00
06AD BD   JMS E1D1      OUT EE
06AE E1
06AF D1
06B0 39   RTS

END
```

```
A002            LOADDR      Port Address MSB
A003                        Port Address LSB

                    Start Loading Program at A048

A048    A0                  Program Counter MSB
A049    4A                  Program Counter LSB
A04A    FE      START       LDX     LOADDR
A04B    A0
A04C    02
A04D    86                  LDA A   #$13
A04E    13
A04F    A7                  STA A   0,X
A050    00
A051    86                  LDA A   #$11
A052    11
A053    A7                  STA A   0,X
A054    00
A055    86      LOOP1       LDA A   #$01
A056    01
A057    A4                  AND A   0,X
A058    00
A059    27                  BEQ     LOOP1
A05A    FA
A05B    86                  LDA A   #$B0
A05C    B0
A05D    A4                  AND A   0,X
A05E    00
A05F    27                  BEQ     SKIP1
A060    06
A061    E6                  LDA B   1,X
A062    01
A063    C6                  LDA B   #$5E
A064    5E
A065    20                  BRA     LOOP2
A066    02
A067    E6      SKIP1       LDA B   1,X
A068    01
A069    86      LOOP2       LDA A   #$02
A06A    02
A06B    A4                  AND A   0,X
A06C    00
A06D    27                  BEQ     LOOP2
A06E    FA
A06F    E7                  STA B   1,X
A070    01
A071    20                  BRA     LOOP1
A072    E2

                        -END-
```

```
A002            LOTEMP      Starting Address MSB
A003                        Starting Address LSB
A004            HITEMP      Ending Address MSB
A005                        Ending Address LSB

                    Start Loading Program at A014

A014  00        INXMSB
A015  00        INXLSB
A016  00        ACCA
A017  2B        FLAG
A018  FE        START   LDX     LOTEMP
A019  A0
A01A  02
A01B  86        LODREG  LDA A   #1
A01C  01
A01D  A7                STA A   0,X
A01E  00
A01F  A1                CMP A   0,X
A020  00
A021  26                BNE     ERRPNT
A022  0D
A023  48        LOOP1   ASL A
A024  68                ASL     0,X
A025  00
A026  A1                CMP A   0,X
A027  00
A028  26                BNE     ERRPNT
A029  06
A02A  81                CMP A   #$80
A02B  80
A02C  26                BNE     LOOP1
A02D  F5
A02E  20                BRA     INCR1
A02F  3B
A030  FF        ERRPNT  STX     INXMSB
A031  A0
A032  14
A033  CE                LDX     #MCL
A034  E1
A035  9D
A036  20                BRA     SKIP1
A037  12

                    Continue Loading Program at A048

A048  A0                        Program Counter MSB
A049  18                        Program Counter LSB
A04A  B7        SKIP 1  STA A   ACCA
```

```
A04B    A0
A04C    16
A04D    BD              JSR     PDATA1
A04E    E0
A04F    7E
A050    CE              LDX     #INXMSB
A051    A0
A052    14
A053    BD              JSR     OUT4HS
A054    E0
A055    C8
A056    CE              LDX     #ACCA
A057    A0
A058    16
A059    BD              JSR     OUT2HS
A05A    E0
A05B    CA
A05C    FE              LDX     INXMSB
A05D    A0
A05E    14
A05F    BD              JSR     OUT2HS
A060    E0
A061    CA
A062    CE              LDX     #MCL
A063    E1
A064    9D
A065    BD              JSR     PDATA1
A066    E0
A067    7E
A068    FE              LDX     INXMSB
A069    A0
A06A    14
A06B    BC      INCR1   CPX     HITEMP
A06C    A0
A06D    04
A06E    27              BEQ     FINISH
A06F    03
A070    08              INX
A071    20              BRA     LODREG
A072    A8
A073    B6      FINISH  LDA A   FLAG
A074    A0
A075    17
A076    BD              JSR     OUTEEE
A077    E1
A078    D1
A079    20              BRA     START
A07A    9D

                END
```

SWTPC 6800 Short Memory Address Convergence MEMCON-1

This Memory Convergence diagnostic is one designed to check for and locate address convergence problems in the SWTPC 6800 Computer System memory boards, MP-M/MP-MX. The program itself uses 56_{10} words and is meant to be loaded within the 128 word RAM used by the MIKBUG® operating system on the MP-A Microprocessor/System board; making the program independent of the MP-M RAM memory. The diagnostic may be loaded from either tape of from the terminal instruction by instruction using MIKBUG® starting from address $A\emptyset14_{16}$ thru $A\emptyset34_{16}$ and then from address $A\emptyset48_{16}$ thru $A\emptyset5E_{16}$. The program must be loaded in two parts to avoid interfering with the system's push-down stack. The section of memory to be tested is set by loading the most significant byte of the lower memory address into $A\emptyset\emptyset2_{16}$, the least significant byte of the lower memory address into $A\emptyset\emptyset3_{16}$, the most significant byte of the upper memory address into $A\emptyset\emptyset4_{16}$, and the least significant byte of the upper memory address into $A\emptyset\emptyset5_{16}$ using MIKBUG® just as is done for MIKBUG® punch routine. The lower and upper addresses are inclusive and may be any addresses between $\emptyset\emptyset\emptyset\emptyset_{16}$ and $FFFF_{16}$ with the only requirement that the lower address be less than or equal to the upper address. Since addresses $A\emptyset5F_{16}$ thru $A\emptyset7F_{16}$ of the MIKBUG® RAM are still available for program use, the diagnostic may run on these locations just to make sure the diagnostic itself is functioning correctly. Since the program counter is set when the program is initially loaded, the routine is initiated after loading according to the "Go To User's Program" section of the Engineering Note 100 in the Operating System section of this notebook. Once initiated, the program can be stopped only by depressing the "RESET" button. The program may then be re-started after setting the program counter to $A\emptyset15_{16}$ at A048 and A049 as described in the Display Contents of MPU Registers Function" section of the Engineering Note 100.

The test sequence starts by loading a continuous stream of 256 sequential binary numbers from the low memory address to the high memory address, inclusive. It then goes back and sequentially reads the data in each of the locations and compares it to what actually should be there. If it finds any discrepancies within the memory cycle, one X is printed and the cycle is re-started, otherwise a # is printed to indicate successful cycle completion. Since the actual location of any detected errors does not point to the source of the problem, no provision is made for indicating the addresses of detected errors. It must also be noted that the program is not 100% effective. It would be possible to set bits in multiple locations that coincidentally would have been set anyway. However, each cycle puts different data in each memory location, so the changes of a missed problem are reduced. The program loops forever and may be exited when desired by depressing the "RESET" switch which loads the MIKBUG® control program.

If you wish to eliminate the cyclic printout of the "#" sign you can do so by changing the data in address locations A059, A05A and A05B to NOP instructions ($\emptyset1_{16}$) using MIKBUG®. This way you only get a printout of the error cycles, if any.

Mikbug® is a registered trademark of Motorola Inc.

```
A002            LOMEM       Starting Address MSB
A003                        Starting Address LSB
A004            HIMEM       Ending Address MSB
A005                        Ending Address LSB

                    Start Loading Program at A014

A014    00      BSTORE
A015    F7      START       STA B   BSTORE
A016    A0
A017    14
A018    FE                  LDX     LOMEM
A019    A0
A01A    02
A01B    E7      LOOP1       STA B   0,X
A01C    00
A01D    BC                  CPX     HIMEM
A01E    A0
A01F    04
A020    27                  BEQ     CHECK
A021    04
A022    08                  INX
A023    5C                  INC B
A024    20                  BRA     LOOP1
A025    F5
A026    F6      CHECK       LDA B   BSTORE
A027    A0
A028    14
A029    FE                  LDX     LOMEM
A02A    A0
A02B    02
A02C    E1      LOOP2       CMP B   0,X
A02D    00
A02E    26                  BNE     ERROR
A02F    20
A030    BC                  CPX     HIMEM
A031    A0
A032    04
A033    20                  BRA     JUMP
A034    15

                  Continue Loading Program at A048

A048    A0
A049    15
A04A    27      JUMP        BEQ     CYCLE
A04B    0B
A04C    08                  INX
```

```
A04D    5C              INC B
A04E    20              BRA     LOOP2
A04F    DC
A050    86      ERROR   LDA A   #'X
A051    58
A052    BD              JSR     OUTEEE
A053    E1
A054    D1
A055    20              BRA     START
A056    BE
A057    86      CYCLE   LDA A   #'#
A058    23
A059    BD              JSR     OUTEEE
A05A    E1
A05B    D1
A05C    5A              DEC B
A05D    20              BRA     START
A05E    B6
```

- END -

Memory Address Assignment Table (Hex)

Board #	Memory Quadrant (K of memory)	Starting Addr.	Ending Addr.
Ø	1	ØØØØ	Ø3FF
	2	Ø4ØØ	Ø7FF
	3	Ø8ØØ	ØBFF
	4	ØCØØ	ØFFF
1	1	1ØØØ	13FF
	2	14ØØ	17FF
	3	18ØØ	1BFF
	4	1CØØ	1FFF
2	1	2ØØØ	23FF
	2	24ØØ	27FF
	3	28ØØ	2BFF
	4	2CØØ	2FFF
3	1	3ØØØ	33FF
	2	34ØØ	37FF
	3	38ØØ	3BFF
	4	3CØØ	3FFF
4	1	4ØØØ	43FF
	2	44ØØ	47FF
	3	48ØØ	4BFF
	4	4CØØ	4FFF
5	1	5ØØØ	53FF
	2	54ØØ	57FF
	3	58ØØ	5BFF
	4	5CØØ	5FFF
6	1	6ØØØ	63FF
	2	64ØØ	67FF
	3	68ØØ	6BFF
	4	6CØØ	6FFF
7	1	7ØØØ	73FF
	2	74ØØ	77FF
	3	78ØØ	7BFF
	4	7CØØ	7FFF

MP-M/MP-MX Memory IC Assignment Map

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit Ø
Quadrant 1 (1K)	IC15	IC13	IC11	IC9	IC7	IC5	IC3	IC1
Quadrant 2 (2K)	IC16	IC14	IC12	IC10	IC8	IC6	IC4	IC2
Quadrant 3 (3K)	IC40	IC38	IC36	IC34	IC32	IC30	IC28	IC26
Quadrant 4 (4K)	IC39	IC37	IC35	IC33	IC31	IC29	IC27	IC25

ØØ hex = ØØØØ ØØØØ binary
Ø1 hex = ØØØØ ØØØ1 binary
Ø2 hex = ØØØØ ØØ1Ø binary
Ø4 hex = ØØØØ Ø1ØØ binary
Ø8 hex = ØØØØ 1ØØØ binary
1Ø hex = ØØØ1 ØØØØ binary
2Ø hex = ØØ1Ø ØØØØ binary
4Ø hex = Ø1ØØ ØØØØ binary
8Ø hex = 1ØØØ ØØØØ binary

SWTPC 6800 Serial Interface Diagnostic SERINT-1

This serial interface diagnostic is one designed to locate problems in the SWTPC 6800 Computer System serial interface board, MP-S. It is assumed that before loading this program the rest of the system is functioning normally with no problems. The program itself uses 41_{10} words and is loaded within the 128 word RAM used by the MIKBUG® operating system on the MP-A Microprocessor/System Board. A program may reside in external RAM memory simultaneously with the diagnostic loaded within the 128 word RAM, or the diagnostic may be run with no MP-M memory boards installed on the system at all. The diagnostic may be loaded either from tape or instruction by instruction using MIKBUG® starting from address AØ48, thru AØ72. The address of the serial interface to be diagnosed is set by using MIKBUG® to load the hexadecimal address of the selected port into memory locations AØØ2 and AØØ3 with the most significant byte going into AØØ2 and the least significant byte going into AØØ3. The starting address locations of the interface ports are given below:

Port	Address in Hex	
I/O #Ø	8ØØØ	
I/O #1	8ØØ4	(reserved for control interface)
I/O #2	8ØØ8	
I/O #3	8ØØC	
I/O #4	8Ø1Ø	
I/O #5	8Ø14	
I/O #6	8Ø18	
I/O #7	8Ø1C	

Since the program counter is set when the program is initially loaded, the diagnostic is initiated as described in the "Go to User's Program" section of the Engineering Note 100. Once initiated, the program can be stopped only by depressing the "RESET" button. The program may then be re-started after resetting the program counter to AØ4A as described in the "Display contents of MPU Registers Function" section of Engineering Note 100.

The diagnostic itself simply programs the selected serial interface to echo all incoming data back to the transmitting device. It does not check all of the internal operations of the interface. Interrupts and parity are not in any way tested, however, it is unlikely that the rest of the interface would check properly with only these portions inoperative. With the diagnostic loaded, the interface may be tested with any serial terminal including, if you are careful, the control terminal which is normally plugged onto the control interface.

To test a selected interface, first attach the baud rate jumper on the interface board for the selected baud rate. With the power off plug the board onto the selected interface position. If you are using a terminal other than the control terminal for testing connect it to the interface's input/output connector along the top edge of the board. Power up the system and load in the diagnostic program and the address of the serial interface to be tested. Then execute a "Go to User's Program" function as described in Engineering Note 100. If you plan to use the control terminal to check the interface, carefully

unplug its connector from the control interface and plug it onto the serial interface under test. You cannot of course remove power from the unit during this procedure since doing so will wipe out the diagnostic program previously loaded into memory. You should make sure also that the baud rate setting on the terminal is the same as that on the interface under test.

At this time you should be able to type in data from the keyboard and have it echoed back by the computer for confirmation of proper interface operation. To return to terminal control, when testing with the control terminal, the control terminal's I/O connector will have to be removed from the serial interface and plugged back onto the control interface where it normally resides.

Never install or remove the interface board when the system is powered up. Doing so is not only hazardous to the equipment, but bypasses the normal power-up sequence required by the internal registers within the 6850 integrated circuit in order to guarantee proper operation.

Motorola's M6800 User's Group

Motorola Inc. has invited SWTPC 6800 Computer System customers to join their M6800 User's Group. The library contains a growing number of programs and subroutines which may be directly or easily modified to run on the SWTPC 6800 Computer System. The reason for possible modification is that some of the programs within the library have been written using character input/output subroutines resident in Motorola's EXOR cisor® firmware. Similar character input/output subroutines are resident in the SWTPC 6800 mini-operating system ROM as well but are located at different addresses.

For those programmers using machine language, you must submit your program using the assembler mnemonics, hand generating the machine code as you go long. Programs written in machine language with no accompanying source statements will not be accepted. It is suggested that all programs you write either for your own use or for submission to Motorola's program library have the first 20_{16} memory locations (ØØØØ thru ØØ1F) unallocated for your program use. We would like to reserve these locations for upward compatibility on future floppy disk file systems.

Be sure to document any submitted program with lots of source statement comments to make it easier for other programmers to understand what you have done.

The M6800 Microprocessor User Group was formed to promote the sharing of information among users of Motorola 6800 microprocessor products. This information takes two forms; the User Group Library and the User Group Newsletter.

USER GROUP LIBRARY

This repository of M6800 programs is available to all User Group members. By adapting programs from the library, users may avoid the time and expense of reinventing the same software "wheels". Program distribution includes a ring binder with source listings of most library programs. The standard distribution includes only descriptions of long programs (listings over 5 pages). Source listings for these programs are available on request for a nominal service charge.

USER GROUP NEWSLETTER

This publication will include information about new library entries, new Motorola microprocessor hardware and software products, and other items of general interest.

User Group membership includes library updates and Newsletters for a period of two years.

MEMBERSHIP REQUIREMENTS

A free two year membership is available to anyone who submits an acceptable entry to the Program Library. Programs should be of general interest, and they must

be documented on the M6800 User Group Library Submittal Form. Programs of all kinds are needed; the following are only suggestions.

- Floating Point Arithmetic Package
- Decimal Arithmetic Package
- Diagnostic Programs
- I/O Drivers
- Desk Calculator Package
- Math Library Routines (squareroot, trig functions, etc.)
- Games
- Etc.

Although users are encouraged to join the User Group by submitting a program, membership may also be purchased for $100.00.

PROGRAMS PRESENTLY IN THE M6800 USER GROUP LIBRARY

1. EXBUG ROUTINE CBCDHX (CONVERT HEX TO BINARY)
2. EXBUG ROUTINE INCHNP (INPUT CHARACTER NO PARITY)
3. EXBUG ROUTINE INCH (INPUT CHARACTER)
4. EXBUG ROUTINE OUTCH (OUTPUT CHARACTER)
5. EXBUG ROUTINE PDATA (PRINT DATA STRING)
6. BINARY TO DECIMAL ASCII CONVERSION
7. HIGH SPEED DOUBLE PRECISION MULTIPLY
8. REENTRANT 16 BIT DIVIDE
9. REENTRANT DOUBLE PRECISION MULTIPLY
10. M6800 RESIDENT I/O PACKAGE - EXBUG VERSION
11. M6800 RESIDENT I/O PACKAGE - MIKBUG VERSION
12. MPU INSTRUCTION TEST
13. MEMORY TEST
14. PIA TEST
15. 32 BIT REENTRANT FLOATING POINT MULTIPLY
16. HIGH DENSITY TAPE LOAD
17. HIGH DENSITY TAPE PUNCH

18. MANUAL PUNCH MODIFICATION FOR M6800 RESIDENT I/O PACKAGE - EXBUG VERSION

19. INTERDATA CROSS ASSEMBLER

20. TIC-TAC-TOE

INSTRUCTIONS

1. Please complete Program Submittal form as follows: (Please print or type)

   a. Author, employer, address, and telephone

   b. Program Type (check appropriate box)

   c. Program Title: Name or brief description of program

   d. Function: Description of the program function and operation

   e. Parameters: Description of register values, memory areas or values accepted as input

   f. Results: Values to be expected in registers, memory areas or as output

   g. Hardware Configuration:

      For Example: SWTPC 6800 Computer System
      Serial Control Interface at 8ØØ4 - 8ØØ7
      Parallel interface at.....

   h. Memory Requirements

   i. Software Support:

      For Example: Mikbug ® ROM MCM6830L7
      Resident software I/O routines

   j. Assembler/Compiler:

      For Example: Machine Language or
      SWTPC Resident Editor/Assembler

2. A source listing (assembler or compiler output) of the program must be included.

3. A test program which can be used for verification of the program must be included.

4. Since your programs may be copied for publication in the User Group Library, please send a clear, unmarked original.

MAIL TO: MOTOROLA INCORPORATED, SUPPORT PRODUCTS, M6800 USER GROUP LIBRARY BB102, P. O. BOX 2953, PHOENIX, ARIZONA 85036

M6800 USER GROUP LIBRARY SUBMITTAL FORM

PROGRAMMER: _____ ADDRESS: _____

COMPANY: _____ _____

TELEPHONE: (OPTIONAL) _____

☐ M6800 RESIDENT ☐ CROSS-COMPUTER

PROGRAM TITLE:

FUNCTION:

PARAMETERS:

RESULTS:

HARDWARE CONFIGURATION: SWTPC 6800 Computer System

MEMORY REQUIRED:

SOFTWARE SUPPORT:

ASSEMBLER/COMPILER:

6800 Software

This section contains a small compilation of programs written for the SWTPC 6800 Computer System. At the time of this writing the programs are all diagnostics intended for checking the various computer system modules for proper operation.

Also contained within this section are details for becoming a member of Motorola's M6800 User's Group. It is Suggested that all programs you write either for your own use or for submission to Motorola's program library have the first 20_{16} memory locations (ØØØØ thur ØØ1F) unallocated for your program use. We would like to reserve these locations for upward compatibility on future floppy disk file systems.

SWTPC 6800 Rotating Bit RAM Memory Diagnostic ROBIT-1

This rotating bit memory diagnostic is designed to check for and locate memory retaining problems in the SWTPC 6800 Computer System memory boards, MP-M/MP-MX. The program itself uses 85_{10} words and is meant to be loaded within the 128 word RAM used by the MIKBUG® operating system on the MP-A Microprocessor/System board. This makes the program independent of the MP-M/MP-MX RAM memory. The diagnostic may be loaded from either tape or from the terminal instruction by instruction using MIKBUG® starting from address $A\emptyset14_{16}$ thru $A\emptyset7A_{16}$. The program must be loaded in two parts to avoid interfering with the sytstem's push-down stack. The contiguous section of memory to be tested is set by loading the most significant byte of the lower memory address into $A\emptyset\emptyset2_{16}$, the least significant byte of the lower memory address into $A\emptyset\emptyset3_{16}$, the most significant byte of the upper memory address into $A\emptyset\emptyset4_{16}$, and the least significant byte of the upper memory address into $A\emptyset\emptyset5_{16}$ using MIKBUG® just as is done for MIKBUG® punch routine. The lower and upper addresses are inclusive and may be any addresses between $\emptyset\emptyset\emptyset\emptyset_{16}$ and $FFFF_{16}$ with the only requirement that the lower address be less than or equal to the upper address. Since addresses $A\emptyset7B_{16}$ thru $A\emptyset7F_{16}$ of the MIKBUG® RAM are still available for program use, the diagnostic may be run on these locations just to make sure the diagnostic itself is functioning correctly. Since the program counter is set when the program is initially loaded, the routine is initated after loading according to the "Go To User's Program" section of the Engineering Note 100 in the Operating System section of this notebook. Once initiated, the program may then be re-started after setting the program counter to $A\emptyset18_{16}$ at A048 and A049 as described in the "Display Contents of MPU Registers Function" section of the Engineering Note 100.

The test sequence starts from the lower address and loads that address with a binary ØØØØ ØØØ1 or $\emptyset1_{16}$. The data in this location is then read and verified. If accurate the "one" bit is shifted left to form a binary ØØØØ ØØ1Ø or $\emptyset2_{16}$ and is then again tested. This shift left sequence continues until a binary 1ØØØ ØØØØ or $8\emptyset_{16}$ has been loaded and verified, at which time the entire sequence is repeated at the next sequential memory address. This sequence continues until the selected upper memory address is reached. The program then prints a "+" on the control terminal to indicate cycle completion and proceeds to repeat itself. The program loops forever and may be exited when desired by depressing the "RESET" switch which loads the MIKBUG® control program. When an error is detected, the memory address followed by what data should have been followed by what the memory data was, are printed out on the control terminal in hexadecimal (base 16) form. Example:

*Ø11Ø Ø2 ØØ

When converted to binary this means that when address Ø11Ø, which is located in the first 1,024 words of RAM memory, was loaded with a binary ØØØØ ØØ1Ø it was read back as containing a binary ØØØØ ØØØØ which indicates a possible problem in the 2^1 bit memory chip in the lower 1,024 words of memory or a possible problem in the 2^1 bit of the memory board data transceiver or a variety of other possibilities. The best way to tell for sure is to look for a pattern in the indicated errors. Take note that once one bit error has been located at a specific memory address, the one error is printed in the form shown above and the program increments to the next address without searching for more errors in the already defective address.

If you wish to eliminate the cyclic printout of the "+" sign you can

do so by changing the data in address locations AØ76, AØ77 and AØ78 to NOP instructions ($Ø1_{16}$) using MIKBUG®. This way you only get a printout of the error locations; that is if there are any. The running time of this program is very fast. It will cycle thru 2,048 words of memory in less than one second.

Mikbug® is a registered trademark of Motorola Inc.